

Question # 1

Query 2 is correct

Reason: This query accurately sets up the Activity table with actID as the primary key and excludes inappropriate foreign keys, aligning with standard database design principles for clarity and integrity.

Question # 2

correct query is **Query 2**

Reason: It uses COUNT(DISTINCT session) to count unique sessions in the 'Main Pool' and properly joins Session and Activity tables on actID.

Question # 3

Query 3 is correct

Reason: It properly joins the Member, Booking, Activity, and Session tables using relevant keys to fetch distinct member names, activity names, and session dates.

Question # 4

Query 3 is Correct

Reason: It accurately joins the Member and Booking tables to fetch the names and emails of members booked for session '123', using the correct key relationships.

Question # 5

Option 1 is correct.

Reason: It directly counts all activities that occur in the "Main Pool" by filtering the Activity table on actLocation.

Question # 6

Query 4 is correct

Reason: It correctly joins the Activity, Session, Booking, and Member tables to list activity names and session dates for sessions booked by 'John Smith,' ensuring that the activity ID, session ID, and member ID correctly match across the tables.

Question # 7

Query 3 or 5 because both are same

Reason: depends on the actual column name , It accurately joins the Activity and Session tables to filter activities by date ('03-JAN-2011') and location ('Main Pool'), matching activity IDs between tables.

Question # 8

Query 2 is correct.

Reason: It directly aggregates prices within the Price table to find the maximum session price for each member category, using GROUP BY to segment by member category.

Question # 9

Query 4 is correct.

Reason: It correctly joins Booking and Session tables, groups by session ID and date, and filters groups having at least 10 bookings using HAVING.

Question # 10

Option 4 is correct.

Reason: depends on the actual column name, It correctly joins Booking and Session tables, groups by session ID and date, and filters groups having at least 10 bookings using HAVING.

Question # 11

Query 2 is the correct

Reason :It directly counts all rows in the Activity table to give the total number of activities.

Question # 12

Query 4 is correct

Reason: It joins Activity, Price, and Member tables to find the session price for a specific activity ('circuits') for a member with a specific ID (00045), using the member category to match the price.

Question # 13

Query 2 is the correct

Reason: It specifically updates the location of the activity named 'baby swim' to 'Baby Pool', directly without affecting others.

Question # 14

Query 1 is correct

Reason: It joins Activity and Session tables, and orders the results by activity name in ascending order and session date in descending order.